모던 웹 애플리케이션 개발2

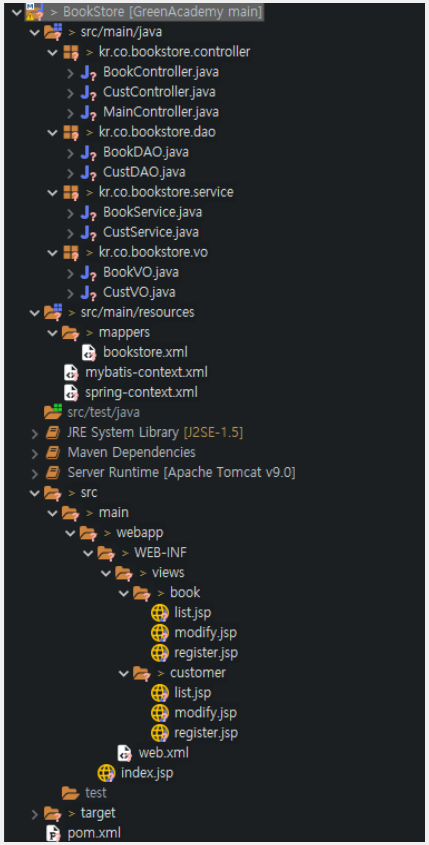
정기 수행 평가 10

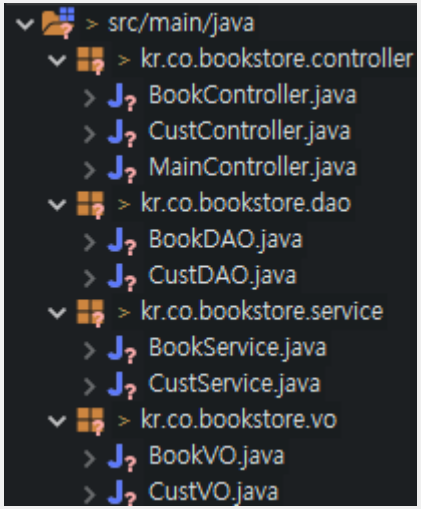
2023.01.04 김동근

Contents

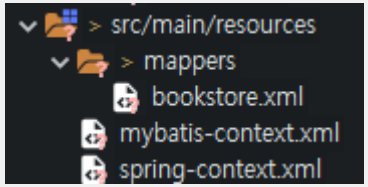
1 프로젝트 구조

| 프로젝트 생성 및 구성

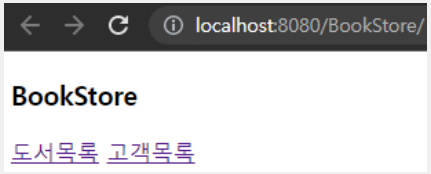




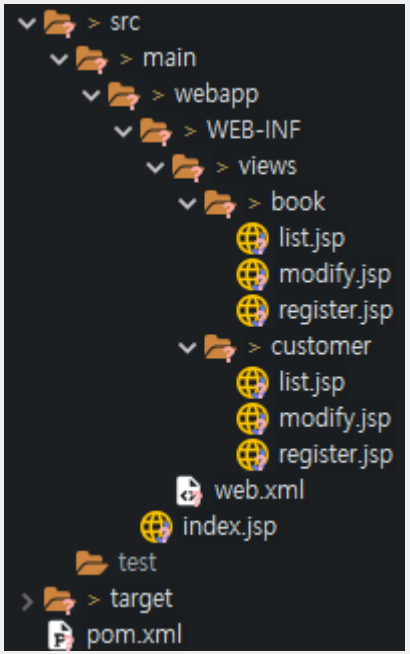
컨트롤러, DAO, Service, VO



DB 및 프로젝트 설정



Index.jsp



화면 및 라이브러리 파일

2 화면 구현

| Book 화면 구현

도서등록

처음으로 도서목록

도서명	
출판사	
가격	
	등록

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠
가격	7001
	수정

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7,001	수정 삭제
2	축구아는 여자	나무수	13,000	수정 삭제
3	Hello World!	GA	89,000	수정 삭제
10	Olympic Champions	Pearson	13,000	수정 삭제

```
<h3>도서목록</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/book/register">도서등록</a>

<table border="1">
<tr>
    <th>도서번호</th>
    <th>도서명</th>
    <th>출판사</th>
    <th>가격</th>
    <th>관리</th>
</tr>
<c:forEach var="book" items="${ books }">
<tr>
    <td>${ book.bookId }</td>
    <td>${ book.bookName }</td>
    <td>${ book.publisher }</td>
    <td><fmt:formatNumber value="${ book.price }" pattern="#,###" /></t
    <td>
        <label><a href="/BookStore/book/modify?id=${ book.bookId }">수정
        <label><a href="/BookStore/book/delete?id=${ book.bookId }">삭제
    </td>
</tr>
</c:forEach>
</table>
```

```
<h3>고객수정</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/customer/list">고객목록</a>
<form action="/BookStore/customer/modify" method="POST">
    <table border="1">
    <tr>
        <td>고객번호</td>
        <td><input type="text" name="custId" value="${ cust.custId }" readonly
    </tr>
    <tr>
        <td>고객명</td>
        <td><input type="text" name="custName" value="${ cust.custName }" /></
    </tr>
    <tr>
        <td>주소</td>
        <td><input type="text" name="custAddr" value="${ cust.custAddr }" /></
    </tr>
    <tr>
        <td>휴대폰</td>
        <td><input type="text" name="custHp" value="${ cust.custHp }" /></td>
    </tr>
    <tr>
        <td colspan="2" align="right"><input type="submit" value="수정" /></td>
    </tr>
    </table>
</form>
```

```
<h3>도서등록</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/book/list">도서목록</a>
<form action="/BookStore/book/register" method="POST">
    <table border="1">
    <tr>
        <td>도서명</td>
        <td><input type="text" name="bookName" /></td>
    </tr>
    <tr>
        <td>출판사</td>
        <td><input type="text" name="publisher" /></td>
    </tr>
    <tr>
        <td>가격</td>
        <td><input type="text" name="price" /></td>
    </tr>
    <tr>
        <td colspan="2" align="right"><input type="submit" value="등록"
    </tr>
    </table>
</form>
```

Book 화면 구현

2 화면 구현

| Customer 화면 구현

고객등록

처음으로 고객목록

고객명	
주소	
휴대폰	
	등록

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 맨체스타
휴대폰	000-5000-0002
	수정

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제

```
<h3>고객목록</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/customer/register">고객등록</a>

<table border="1">
<tr>
    <th>고객번호</th>
    <th>고객명</th>
    <th>주소</th>
    <th>휴대폰</th>
    <th>관리</th>
</tr>
<c:forEach var="cust" items="${ custs }">
<tr>
    <td>${ cust.custId }</td>
    <td>${ cust.custName }</td>
    <td>${ cust.custAddr }</td>
    <td>${ cust.custHp }</td>
    <td>
        <label><a href="/BookStore/customer/modify?id=${ cust.custId }">수정
        <label><a href="/BookStore/customer/delete?id=${ cust.custId }">삭제
    </td>
</tr>
</c:forEach>
</table>
```

```
<h3>고객수정</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/customer/list">고객목록</a>
<form action="/BookStore/customer/modify" method="POST">
    <table border="1">
    <tr>
        <td>고객번호</td>
        <td><input type="text" name="custId" value="${ cust.custId }" readonly
    </tr>
    <tr>
        <td>고객명</td>
        <td><input type="text" name="custName" value="${ cust.custName }" /></t
    </tr>
    <tr>
        <td>주소</td>
        <td><input type="text" name="custAddr" value="${ cust.custAddr }" /></t
    </tr>
    <tr>
        <td>휴대폰</td>
        <td><input type="text" name="custHp" value="${ cust.custHp }" /></td>
    </tr>
    <tr>
        <td colspan="2" align="right"><input type="submit" value="수정" /></td>
    </tr>
    </table>
</form>
```

```
<h3>고객등록</h3>
<a href="/BookStore/">처음으로</a>
<a href="/BookStore/customer/list">고객목록</a>
<form action="/BookStore/customer/register" method="POST">
    <table border="1">
    <tr>
        <td>고객명</td>
        <td><input type="text" name="custName" /></td>
    </tr>
    <tr>
        <td>주소</td>
        <td><input type="text" name="custAddr" /></td>
    </tr>
    <tr>
        <td>휴대폰</td>
        <td><input type="text" name="custHp" /></td>
    </tr>
    <tr>
        <td colspan="2" align="right"><input type="submit" value="등록"
    </tr>
    </table>
</form>
```

3 기능 구현
| CRUD

```java
@Controller
public class BookController {

    @Autowired
    BookService service;

    @GetMapping("/book/list")
    public String list(Model model) {
        List<BookVO> books = service.selectBooks();
        model.addAttribute("books", books);
        return "/book/list";
    }

    @GetMapping("/book/register")
    public String register() {
        return "/book/register";
    }
    @PostMapping("/book/register")
    public String register(BookVO vo) {
        service.insertBook(vo);
        return "redirect:/book/list";
    }

    //modify
    @GetMapping("/book/modify")
    public String modify(Model model, String id) {
        BookVO book = service.selectBook(id);
        model.addAttribute("book", book);
        return "/book/modify";
    }
    @PostMapping("/book/modify")
    public String modify(BookVO vo) {
        service.updateBook(vo);
        return "redirect:/book/list";
    }

    //delete
    @GetMapping("/book/delete")
    public String modify(String id) {
        service.deleteBook(id);
        return "redirect:/book/list";
    }

}
```

```java
@Service
public class BookService {

    @Autowired
    private BookDAO dao;

    public void insertBook(BookVO vo) {
        dao.insertBook(vo);
    }
    public BookVO selectBook(String id) {
        return dao.selectBook(id);
    }
    public List<BookVO> selectBooks() {
        return dao.selectBooks();
    }
    public void updateBook(BookVO vo) {
        dao.updateBook(vo);
    }
    public void deleteBook(String id) {
        dao.deleteBook(id);
    }

}
```

```java
@Repository
public class BookDAO {

    @Autowired
    private SqlSessionTemplate mb;

    public void insertBook(BookVO vo) {
        mb.insert("bookstore.insertBook", vo);
    }
    public BookVO selectBook(String id) {
        return mb.selectOne("bookstore.selectBook", id);
    }
    public List<BookVO> selectBooks() {
        return mb.selectList("bookstore.selectBooks");
    }
    public void updateBook(BookVO vo) {
        mb.update("bookstore.updateBook", vo);
    }
    public void deleteBook(String id) {
        mb.delete("bookstore.deleteBook", id);
    }
}
```

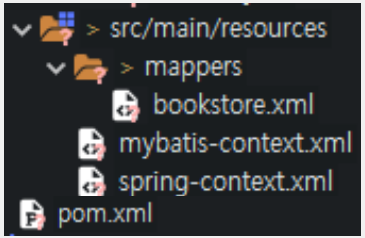
```java
@Controller
public class CustController {

    @Autowired
    CustService service;

    @GetMapping("/customer/list")
    public String list(Model model) {
        List<CustVO> custs = service.selectCusts();
        model.addAttribute("custs", custs);
        return "/customer/list";
    }

    @GetMapping("/customer/register")
    public String register() {
        return "/customer/register";
    }
    @PostMapping("/customer/register")
    public String register(CustVO vo) {
        service.insertCust(vo);
        return "redirect:/customer/list";
    }

    //modify
    @GetMapping("/customer/modify")
    public String modify(Model model, String id) {
        CustVO cust = service.selectCust(id);
        model.addAttribute("cust", cust);
        return "/customer/modify";
    }
    @PostMapping("/customer/modify")
    public String modify(CustVO vo) {
        service.updateCust(vo);
        return "redirect:/customer/list";
    }

    //delete
    @GetMapping("/customer/delete")
    public String modify(String id) {
        service.deleteCust(id);
        return "redirect:/customer/list";
    }

}
```

```java
@Service
public class CustService {

    @Autowired
    private CustDAO dao;

    public void insertCust(CustVO vo) {
        dao.insertCust(vo);
    }
    public CustVO selectCust(String id) {
        return dao.selectCust(id);
    }
    public List<CustVO> selectCusts() {
        return dao.selectCusts();
    }
    public void updateCust(CustVO vo) {
        dao.updateCust(vo);
    }
    public void deleteCust(String id) {
        dao.deleteCust(id);
    }

}
```

```java
@Repository
public class CustDAO {

    @Autowired
    private SqlSessionTemplate mb;

    public void insertCust(CustVO vo) {
        mb.insert("bookstore.insertCust", vo);
    }
    public CustVO selectCust(String id) {
        return mb.selectOne("bookstore.selectCust", id);
    }
    public List<CustVO> selectCusts() {
        return mb.selectList("bookstore.selectCusts");
    }
    public void updateCust(CustVO vo) {
        mb.update("bookstore.updateCust", vo);
    }
    public void deleteCust(String id) {
        mb.delete("bookstore.deleteCust", id);
    }
}
```

Controller, Service, DAO

3 기능 구현
| Mybatis 사용



주요 구조

```xml
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <mappers>
    <mapper resource="./mappers/bookstore.xml"/>
  </mappers>
</configuration>
```

mybatis-context.xml
SQL 문이 있는 mapper 등록

```xml
<!-- https://mvnrepository.com/artifact/org.mybat
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.11</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.mybat
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.1.0</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.24</version>
</dependency>
<!-- https://mvnrepository.com/artifact/mysql/mys
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.31</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apach
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.9.0</version>
</dependency>
```

pom.xml
Mybatis, dbcp
라이브러리 다운

```xml
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        https://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
        https://www.springframework.org/schema/mvc/spring-mvc.xsd"
        >
    <!-- 하위 패키지의 모든 Component 어노테이션을 추적 검색하여 컨테이너에 객체 등록 -->
    <context:component-scan base-package="kr.co.bookstore"/>

    <!-- 스프링 Web MVC 관련 어노테이션을 위한 태그 선언 -->
    <mvc:annotation-driven></mvc:annotation-driven>

    <!-- 스프링 MVC ViewResolver 등록 -->
    <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views/"/>
        <property name="suffix" value=".jsp"/>
    </bean>

    <!-- 데이터베이스(커넥션풀) 설정 -->
    <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />
        <property name="url" value="jdbc:mysql://127.0.0.1:3306/java1_bookstore" />
        <property name="username" value="root" />
        <property name="password" value="1234" />
    </bean>
    <!-- Mybatis 설정 -->
    <bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="configLocation" value="classpath:mybatis-context.xml"></property>
    </bean>

    <!-- Mybatis-Spring 설정 -->
    <bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
        <constructor-arg ref="sqlSessionFactoryBean"></constructor-arg>
    </bean>
</beans>
```

bookstore에 연결

Mybatis 라이브러리 다운 후 클래스 연결

spring-context.xml
풀 생성, Mybatis 객체 등록 설정

3 기능 구현
| Mybatis 사용

```java
public class BookVO {
    private int bookId;
    private String bookName;
    private String publisher;
    private int price;
```

BookVO.java

```java
public class CustVO {
    private int custId;
    private String custName;
    private String custAddr;
    private String custHp;
```

CustVO.java

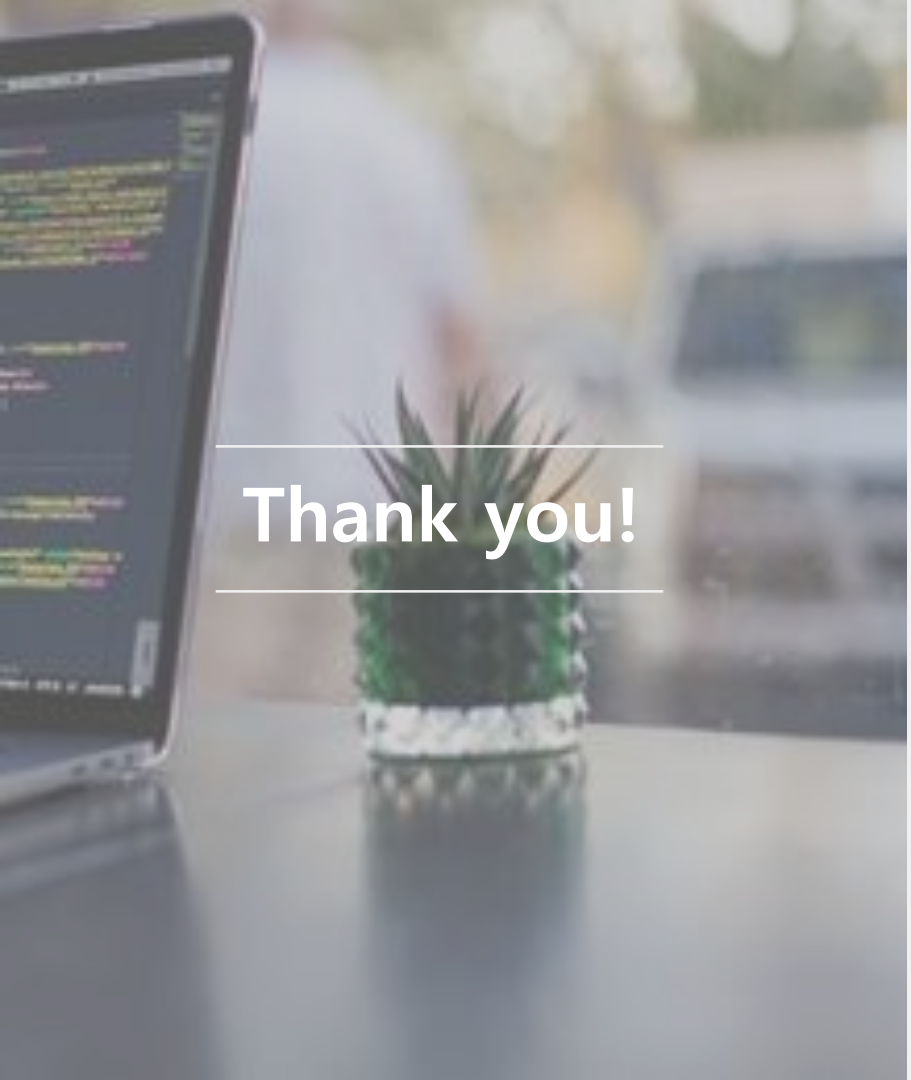
```xml
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="bookstore">

    <!-- book -->
    <insert id="insertBook">
        INSERT INTO `book` SET `bookName`=#{bookName}, `publisher`=#{publisher}, `price`=#{price};
    </insert>

    <select id="selectBook" resultType="kr.co.bookstore.vo.BookVO">
        SELECT * FROM `book` WHERE `bookId`=#{bookId};
    </select>

    <select id="selectBooks" resultType="kr.co.bookstore.vo.BookVO">
        SELECT * FROM `book`;
    </select>

    <update id="updateBook">
        UPDATE `book` SET `bookName`=#{bookName}, `publisher`=#{publisher}, `price`=#{price} WHERE `bookId`=#{bookId};
    </update>

    <delete id="deleteBook">
        DELETE FROM `book` WHERE `bookId`=#{bookId};
    </delete>
<!-- cust -->
<insert id="insertCust">
    INSERT INTO `customer` SET `custName`=#{custName}, `custAddr`=#{custAddr}, `custHp`=#{custHp};
</insert>

<select id="selectCust" resultType="kr.co.bookstore.vo.CustVO">
    SELECT * FROM `customer` WHERE `custId`=#{custId};
</select>

<select id="selectCusts" resultType="kr.co.bookstore.vo.CustVO">
    SELECT * FROM `customer`;
</select>

<update id="updateCust">
    UPDATE `customer` SET `custName`=#{custName}, `custAddr`=#{custAddr}, `custHp`=#{custHp} WHERE `custId`=#{custId};
</update>

<delete id="deleteCust">
    DELETE FROM `customer` WHERE `custId`=#{custId};
</delete>
```

Bookstore.xml
SQL문

4 실행

| 동작 화면

동작 화면



정기 수행 평가

10.모던 웹 애플리케이션 개발2
BookStore

Kim Dong Geun
From GreenAcademy
2023.01.04